

BAB III

PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem merupakan penjabaran dari suatu sistem informasi yang utuh ke dalam berbagai bagian komponennya dengan maksud agar bisa mengidentifikasi dan mengevaluasi berbagai macam masalah atau hambatan yang timbul pada sistem sehingga nantinya bisa dilakukan penanggulangan, perbaikan dan juga pengembangan. Dalam pembangunan suatu aplikasi harus dilakukannya analisis, diantaranya :

3.1.1 Analisis Masalah

Berdasarkan analisis masalah dalam pembuatan aplikasi ini, diantaranya :

1. Minimnya informasi lowongan pekerjaan paruh waktu
2. Sulit mendapatkan pekerja paruh waktu
3. Sulit mendapatkan pekerja yang terdekat dari lokasi kita
4. Tidak terwadahnya para mahasiswa khususnya yang membutuhkan pekerjaan paruh waktu
5. Tidak adanya aplikasi untuk mempermudah para penyedia lowongan pekerjaan memberikan informasi.

3.1.2 Analisis Algoritma

Analisis algoritma dibuat untuk mengetahui cara kerja algoritma dan mengetahui kelebihan dan kekurangan dari aplikasi yang dibangun menggunakan algoritma *haversine*.

3.1.2.1 Cara Kerja Algoritma

Cara kerja algoritma *haversine* pada pembangunan aplikasi ini dapat dijelaskan sebagai berikut :

```

final int R = 6371;

Double lat1 = latitude;

Double lon1 = longitude;

Double lat2 = mListMarker.get(i).getLatitude();

Double lon2 = mListMarker.get(i).getLongitude();

Double latDistance = toRad(lat2-lat1);

Double lonDistance = toRad(lon2-lon1);

Double a = Math.sin(latDistance / 2) * Math.sin(latDistance / 2) +
Math.cos(toRad(lat1)) * Math.cos(toRad(lat2)) * Math.sin(lonDistance / 2)
* Math.sin(lonDistance / 2);

Double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));

Double distance = R * c;

```

Langkah – langkah algoritma :

langkah 1 mengetahui *radius* bumi

langkah 2 mendekteksi *latitude user*

langkah 3 mendeteksi *longitude user*

langkah 4 mengambil data *latitude* semua pegawai yang sudah *signup*

langkah 5 mengambil data *longitude* semua pegawai/yang sudah *signup*

langkah 6 mengambil *latitude* pegawai yang sudah *sign up* dikurangi *latitude* posisi kita/user untuk mendapatkan perhitungan jarak *latitude*

langkah 7 mengambil *longitude* pegawai yang sudah *signup* dikurangi *latitude* kita untuk mendapatkan perhitungan jarak *longitude*

langkah ke 8 menghitung jarak diantara *latitude* dan *longitude* yang sudah didapatkan

langkah ke 9 mendapatkan jarak di antara i dan pegawai

3.1.2.2 Kelebihan dan Kekurangan Algoritma

Adapun kelebihan dan kekurangan algoritma yang digunakan dalam pembangunan aplikasi ini sebagai berikut.

1. Kelebihan Algoritma

Pada penelitian yang dilakukan oleh ingole dan nichat menyampaikan hasil penelitian untuk mencari *route* jalur terpendek menggunakan algoritma *haversine*. Penelitian ini berhasil menerapkan algoritma *haversine* dengan cara *user* menginputkan lokasi awal dan lokasi akhir.

2. Kekurangan Algoritma

Dari hasil penelitian dapat disimpulkan kekurangan dari algoritma *haversine* hanya dapat mengukur jarak antar 2 titik berdasarkan inputan antar 2 aktor dengan cara menginput lokasi awal dan lokasi akhir. Algoritma *haversine* hanya dapat menghitung jarak antara satu koordinat dengan koordinat lain secara garis lurus, dan mengabaikan bukit ataupun lembah yang ada di permukaan, input dari metode ini adalah berupa *latitude* dan *longitude* lokasi yang akan dihitung dan *outputnya* berupa nilai jarak antara kedua lokasi.

3.1.3 Analisis Kebutuhan Data

Analisis kebutuhan data dibuat untuk mengklasifikasikan kebutuhan dalam perancangan aplikasi, yang nantinya akan mempermudah dalam pembangunannya, berikut kebutuhan data dari hasil analisis :

Table 1-1 Analisis kebutuhan data

No	Field Name	Type	Description
----	------------	------	-------------

1	Identitas Diri	<i>Teks</i>	<i>Teks</i> yang digunakan untuk mengetahui informasi tentang <i>employee</i>
2	Lokasi / Posisi	<i>Map</i>	Digunakan untuk mengetahui titik keberadaan dari <i>employee</i>
3	Gambar	<i>Img</i>	Gambar yang digunakan untuk mengetahui wajah <i>employee</i>

3.1.4 Analisis Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang berisi proses-proses apa saja / layanan apa saja yang nantinya harus disediakan oleh sistem agar berjalan sesuai dengan kebutuhan.

3.1.2.3 List Requirements

List Requirement menjelaskan tentang apa saja yang dapat dilakukan oleh aplikasi, yang nantinya akan menjadi bahan pengujian pada aplikasi. Berikut adalah *list requirement* pada aplikasi ini :

Table 1-2 List Requirement

No	Fungsi Aplikasi	Keterangan
1	Menu Utama	Dapat melihat menu utama
2	Menu Utama <i>Employee Seeker</i>	Dapat melihat menu utama <i>employee seeker</i>
3	Menu Utama <i>Employee</i>	Dapat melihat menu utama <i>employee</i>
4	<i>Filtering</i> Kebutuhan	Dapat melakukan <i>filtering</i> karyawan sesuai yang dibutuhkan

5	Pencarian Karyawan	Dapat melakukan pencariin terhadap karyawan
6	Info Biodata <i>Employee</i>	Dapat melihat info biodata <i>employee</i>
7	Hubungi <i>Employee</i>	Dapat berkomunikasi langsung dengan <i>employee</i>
8	<i>Login Employee</i>	Dapat melakukan <i>login</i>
9	Membuat Akun Baru	Dapat melakukan <i>registrasi</i> pada aplikasi
10	<i>Sign Up</i>	Dapat melakukan <i>registrasi</i> pada aplikasi
11	<i>Save Data Employee</i>	Dapat mengisi data-data karyawan
12	Pengelolaan Data <i>Employee</i>	Dapat melihat hasil karyawan
13	<i>Edit Data Employee</i>	Dapat mengubah data-data yang kurang pas
14	<i>Sign Out</i>	Keluar akun

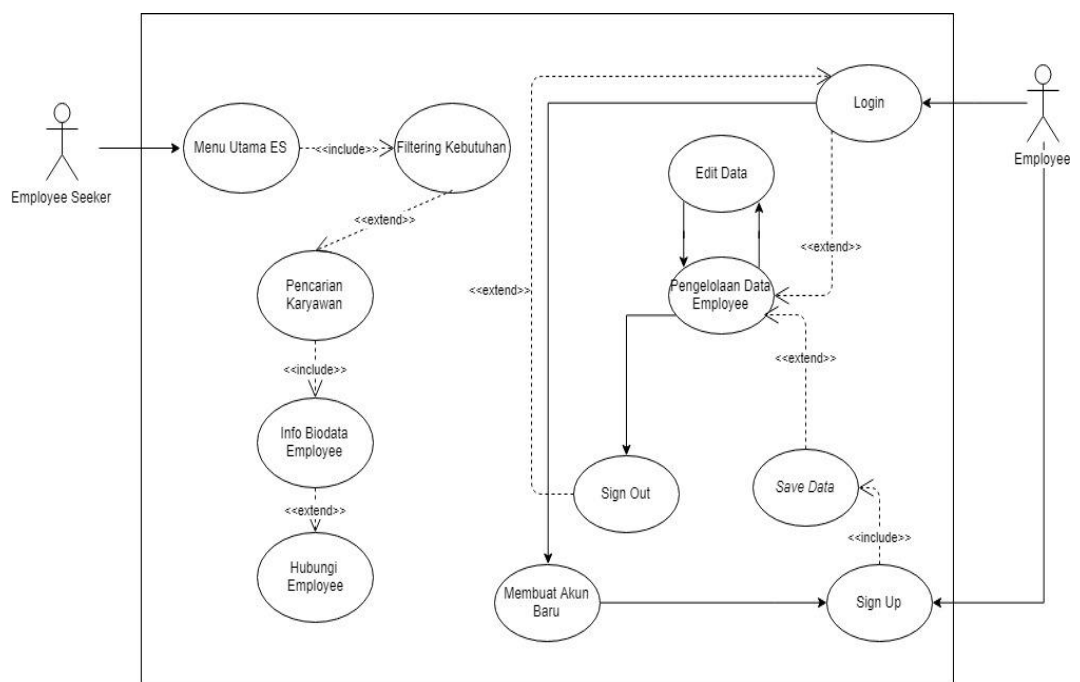
3.1.5 Pemodelan Sistem

Pemodelan sistem dilakukan dengan metode berorientasi obyek dengan menggunakan standarisasi *Unified Modeling Language* (UML). Tahap-tahap pemodelan dalam menggunakan UML antara lain :

1. *Use Case Diagram*
2. *Activity Diagram*
3. *Sequence Diagram*
4. *Collaboration Diagram*
5. *Class Diagram*

3.1.2.4 Use Case Diagram

Use Case merupakan sebuah teknik yang digunakan dalam pengembangan sebuah *software* atau sistem informasi untuk menangkap kebutuhan fungsional dari sistem yang bersangkutan, *Use Case* menjelaskan interaksi yang terjadi antara ‘aktor’ — inisiator dari interaksi sistem itu sendiri dengan sistem yang ada, sebuah *Use Case* direpresentasikan dengan urutan langkah yang sederhana.



Gambar 1-1 Use Case Aplikasi

1. *Employee Seeker* dan *Employee*
Employee Seeker dan *Employee* berperan sebagai user atau aktor yang terdapat pada *use case diagram*. Definisi aktor antara lain :

Table 1-3 *Employee Seeker* dan *Employee*

No	Aktor	Definisi
1	Employee Seeker	Aktor yang mencari employee paruh waktu

2	Employee	Aktor yang mencari pekerjaan paruh waktu
---	----------	--

2. Definisi *Use Case* Aplikasi

Menggambarkan secara garis besar dari sistem yang akan dibangun.

Table 1-4 Use Case Aplikasi

No	Use Case	Definisi
1	Menu Utama ES (<i>Employee Seeker</i>)	Terdapat dua menu didalamnya, ada koki dan tukang kebun, yang nantinya dapat memilih salah satu sesuai yang diinginkan
2	<i>Filtering</i> Kebutuhan	Mengisi data-data <i>employee</i> sesuai kriteria yang diinginkan, mulai dari Jenis kelamin, Waktu mulai kerja, Waktu selesai kerja, dan Gaji per jamnya.
3	Pencarian Karyawan	Setelah mengisi data <i>employee</i> , maka akan keluar jumlah <i>employee</i> dari masing-masing lokasi sesuai kebutuhan yang diinginkan
4	Info Biodata <i>Employee</i>	Ketika selesai memilih, maka akan muncul biodata dan kontak dari <i>employee</i> tersebut dan bisa kita hubungi.
5	Hubungi	<i>Employee seeker</i> dapat menghubungi <i>employee</i>
6	<i>Login</i>	<i>Employee</i> dapat melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> .
7	<i>Sign Up</i>	<i>Employee</i> mendaftarkan <i>username</i> dan <i>password</i>
8	<i>Save Data</i>	<i>Employee</i> mendaftarkan biodata diri

9	Pengelolaan Data <i>Employee</i>	<i>Employee</i> dapat melihat hasilnya apakah sudah sesuai atau belum dan bisa melakukan perubahan data apabila belum sesuai
10	<i>Edit data</i>	<i>Employee</i> dapat mengubah data yang sudah disimpan
11	Membuat Akun Baru	<i>Employee</i> dapat membuat akun baru pada form <i>login</i>
12	<i>Sign Out</i>	<i>Employee</i> dapat keluar dari akun

3.1.2.5 Skenario Use Case Diagram

Setiap *use case* diagram dilengkapi dengan *scenario*, *scenario use case* adalah alur jalannya proses *use case* dari sisi *user* / aktor dan sistem, berikut adalah *skenario*nya :

1. Skenario Use Case Menu Utama *Employee Seeker*

Table 1-5 Menu Utama *Employee Seeker*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Aktor menekan tombol <i>employee seeker</i>	
	2. Sistem menerima permintaan selanjutnya menampilkan menu utama ES

2. Skenario Use Case Filtering Kebutuhan

Table 1-6 Filtering Kebutuhan

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Aktor menekan tombol salah satu menu	
	2. Sistem menampilkan <i>filtering</i> data
3. Aktor memasukkan jenis kelamin, mulai bekerja, selesai bekerja, gaji dan jarak lokasi maksimal karyawan	
	4. Sistem melakukan pencarian sesuai kebutuhan

3. Skenario Use Case Pencarian Karyawan *Employee Seeker*

Table 1-7 Pencarian Karyawan *Employee Seeker*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Aktor menekan tombol salah satu karyawan yang ada	
	2. Sistem menampilkan informasi nama dan kontak
3. Aktor menekan kembali karyawan yang diinginkan	

	4. Sistem memproses permintaan dan menampilkan profile karyawan
--	---

4. *Skenario Use Case Info Biodata Employee*

Table 1-8 Info Biodata Employee

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Aktor memilih karyawan yang diinginkan	
	2. Sistem menampilkan biodata <i>employee</i>

5. *Skenario Use Case Hubungi*

Table 1-9 Hubungi

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Aktor menekan tombol hubungi	
	2. Sistem menampilkan kontak <i>employee</i>

6. *Skenario Use Case Login Employee*

Table 1-10 Login Employee

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Aktor memasukkan <i>username</i> dan <i>password</i> pada layer <i>login</i>	
	2. Sistem melakukan pengecekan <i>validasi username</i> dan <i>password</i> ke <i>database</i>
Skenario Lain	
1. Aktor memasukkan <i>username</i> dan <i>password</i> yang tidak <i>valid</i>	
	2. Sistem menampilkan pesan bahwa <i>username</i> dan <i>password</i> yang dimasukkan tidak terdaftar

7. Skenario Use Case Sign Up Employee

Table 1-11 Sign Up Employee

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Aktor mengisi <i>username</i> , <i>password</i> dan <i>email</i>	

	2. Sistem menyimpan data ke <i>database</i> dan menampilkan <i>layer</i> selanjutnya
--	--

8. *Skenario Use Case Save Data Employee*

Table 1-12 Save Data Employee

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Aktor mengisi semua kebutuhan yang diminta dan menekan tombol save data	
	2. Sistem menyimpan data dan menampilkan hasil pengisian

9. *Skenario Use Case Pengelolaan Data Employee*

Table 1-13 Pengelolaan Data Employee

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Menekan tombol login setelah mengisi username dan password	
	2. Menampilkan biodata employee

10. Skenario Use Case Sign Out Employee

Table 1-14 Sign Out Employee

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Aktor menekan tombol sign out	
	2. Aktor keluar dari akun

11. Skenario Use Case Membuat Akun Baru

Table 1-15 Membuat Akun Baru

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Aktor menekan tombol <i>create new account</i>	
	2. Menampilkan <i>form registrasi</i>

12. Skenario Use Case Edit Data

Table 1-16 Edit Data

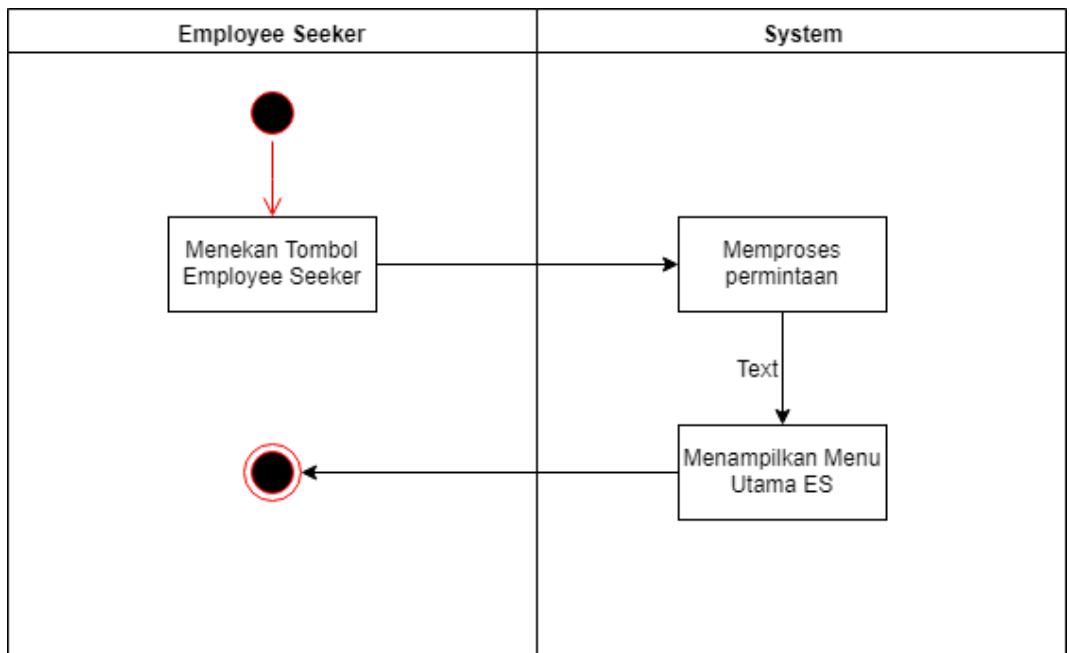
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Aktor menekan pada bagian <i>profile</i>	
	2. Sistem menampilkan data yang ingin diubah
3. Aktor menekan tombol <i>save data</i>	
	4. Data terbaru disimpan

3.1.2.6 Activity Diagram

Activity Diagram merupakan alur kerja (*workflow*) atau kegiatan (aktivitas) dari sebuah sistem atau menu yang ada pada perangkat lunak. *Activity Diagram* juga digunakan untuk mendefinisikan urutan atau pengelompokan tampilan dari sistem /*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan serta rancang menu yang ditampilkan pada perangkat lunak, berikut *beberapa activity diagram* yang ada pada aplikasi dan dapat dilihat pada gambar dibawah ini.

1. Activity Diagram Menu Utama ES

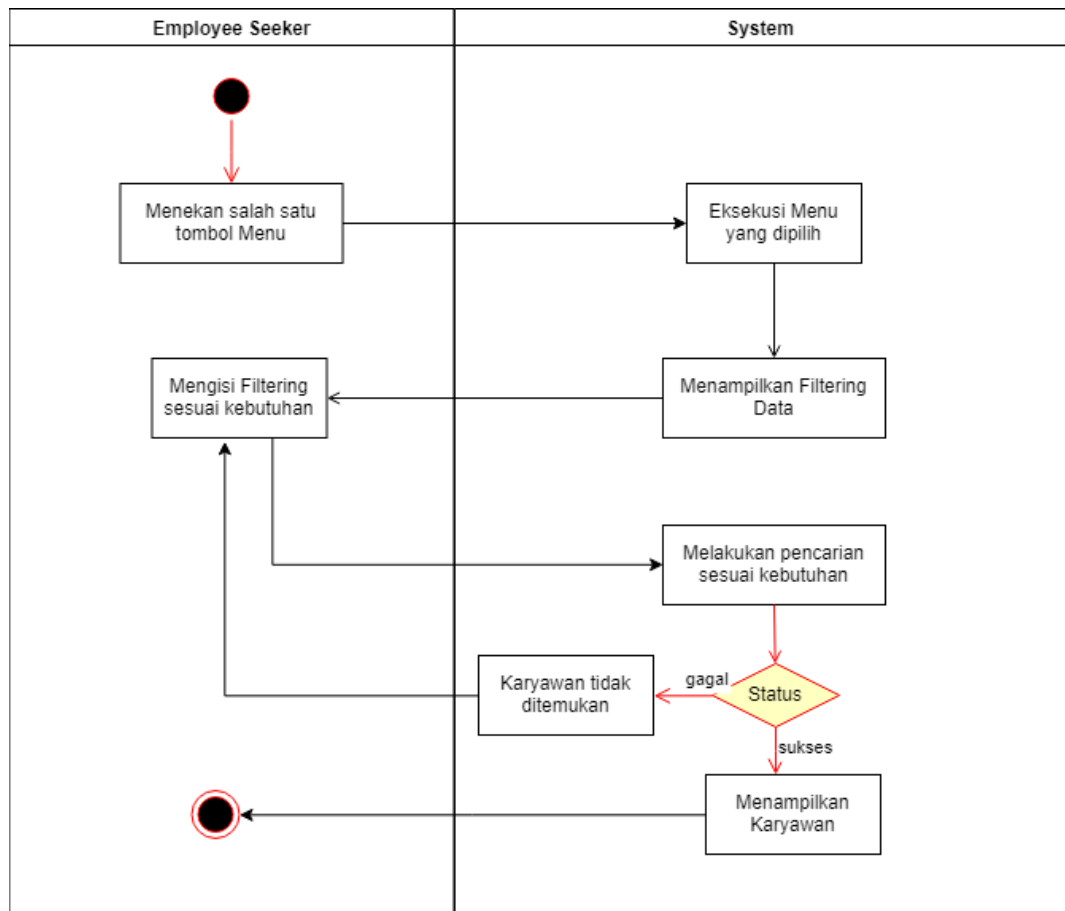
Activity Diagram Menu Utama ES (*Employee Seeker*) yang nantinya aktor diharuskan untuk menekan tombol *employee seeker* terlebih dahulu, lalu akan muncul menu utama dari *Employee Seeker* tersebut, berikut digambarkan seperti dibawah ini.



Gambar 1-2 Menu Utama ES

2. Activity Diagram Filtering Kebutuhan

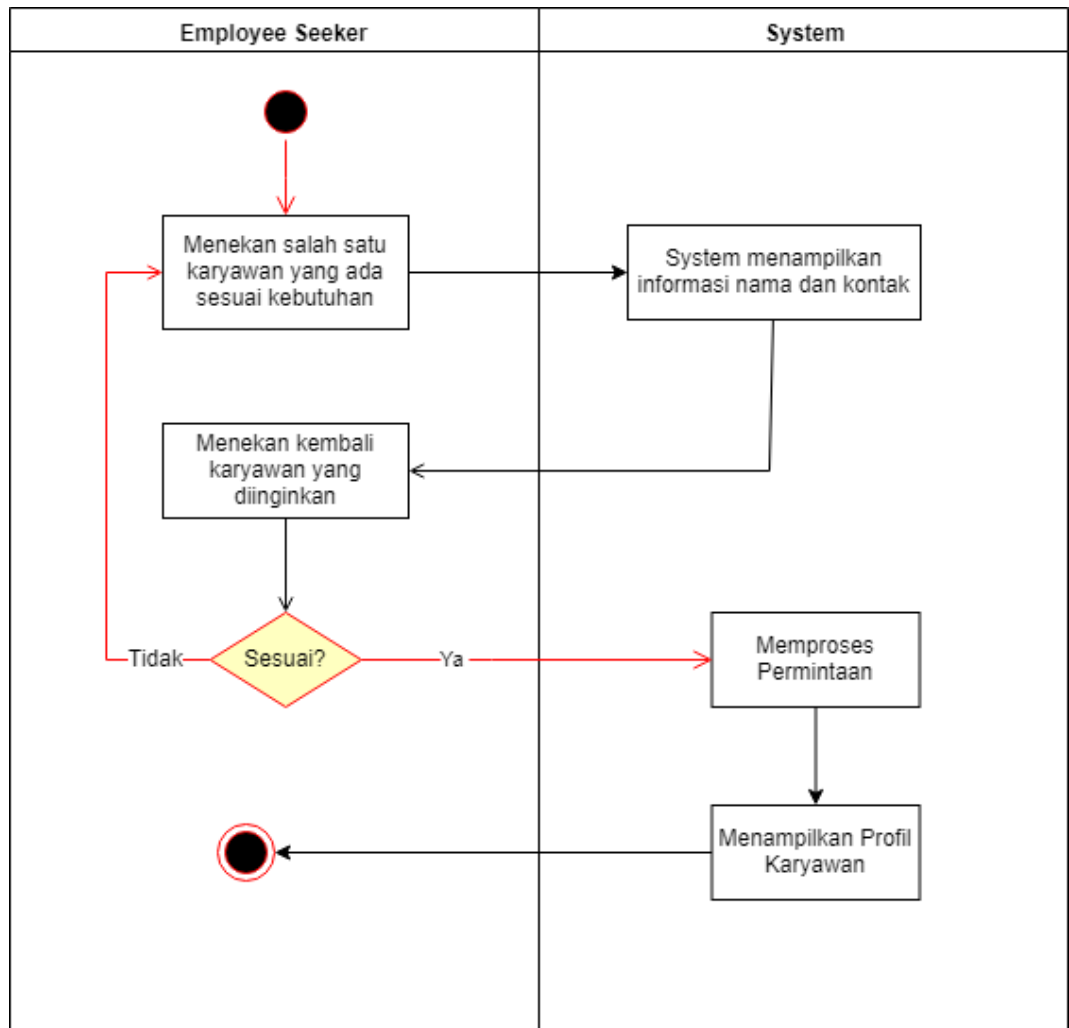
Activity Diagram Filtering kebutuhan yang nantinya aktor diharuskan memilih salah satu menu yang sesuai dengan kebutuhannya, terdapat menu *koki* dan juga tukang kebun, lalu setelah memilih nantinya *employee seeker* diminta untuk mengisi data-data yang dibutuhkan untuk *filtering*, mulai dari Jenis kelamin, Waktu mulai bekerja, Waktu selesai bekerja, dan begitu juga dengan gaji perjam, apabila dirasa data sudah cukup maka dilakukannya pencarian dan akan muncul beberapa karyawan sesuai dengan keinginan, berikut digambarkan seperti dibawah ini.



Gambar 1-3 Filtering Kebutuhan

3. Activity Diagram Pencarian Karyawan

Activity Diagram Pencarian Karyawan yang nantinya setelah mengisi kebutuhan karyawan yang diinginkan kita dapat mulai mencari karyawan yang terdekat dari lokasi kita mulai dari jarak 1 km, 2km, 3km, 4km, dan 5km. Dapat digambarkan seperti dibawah ini.

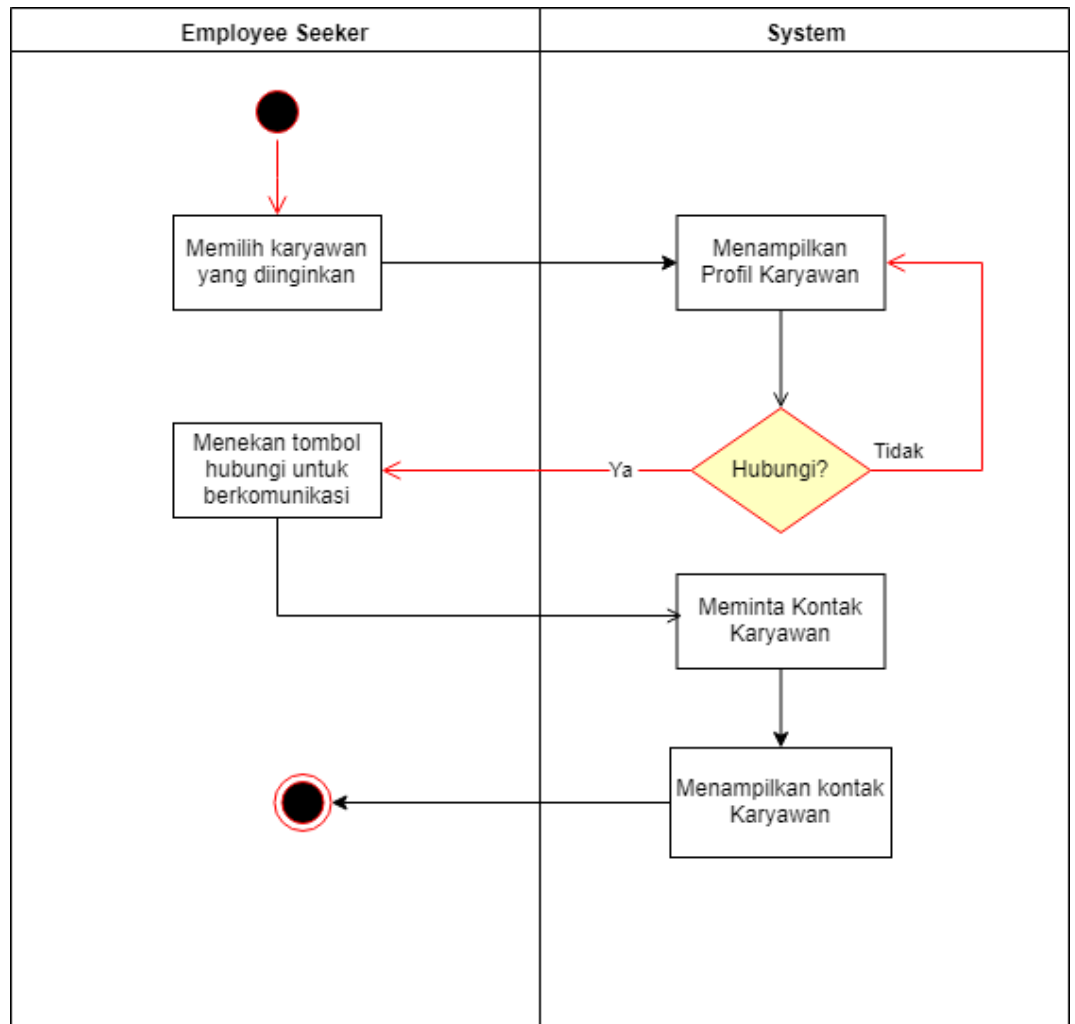


Gambar 1-4 Pencarian Karyawan

4. Activity Diagram Info Biodata Employee

Activity Diagram Info Biodata Employee yang nantinya setelah *employee seeker* memilih dari beberapa karyawan yang tersedia maka akan muncul secara detail info biodata dari *employee* yang kita pilih, disitu juga terdapat pengalaman pekerjaan yang bisa jadi modal penguat

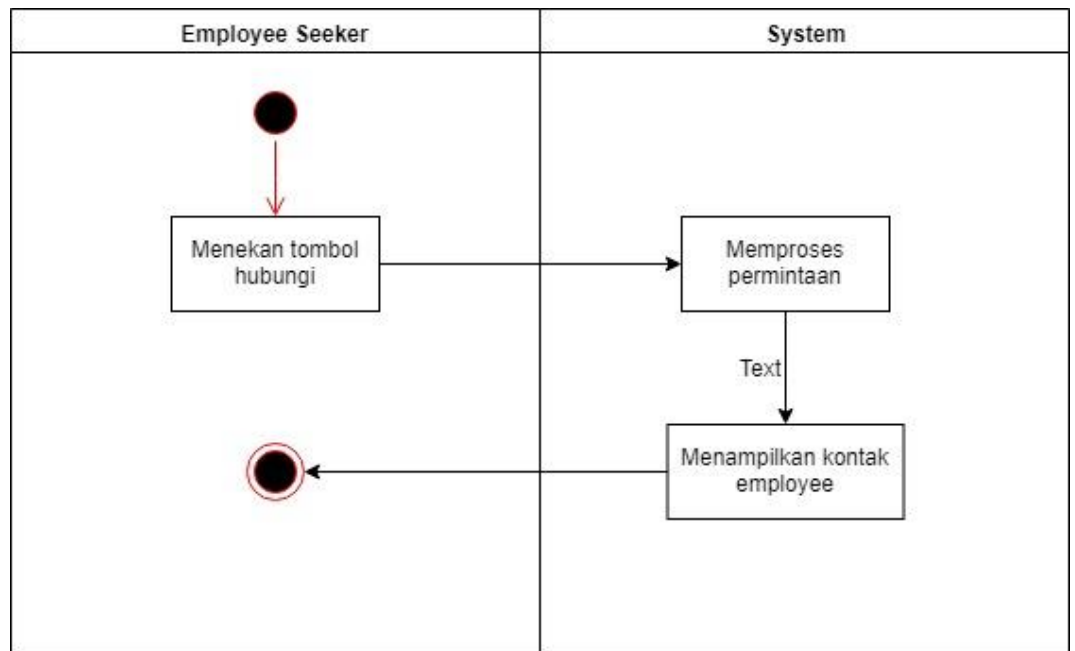
juga untuk kita memilih karyawan tersebut. Apabila dirasa cocok dengan karyawan tersebut kita bisa langsung menghubungi dan melakukan transaksi, dapat digambarkan seperti dibawah ini.



Gambar 1-5 Info Biodata Employee

5. Activity Diagram Hubungi

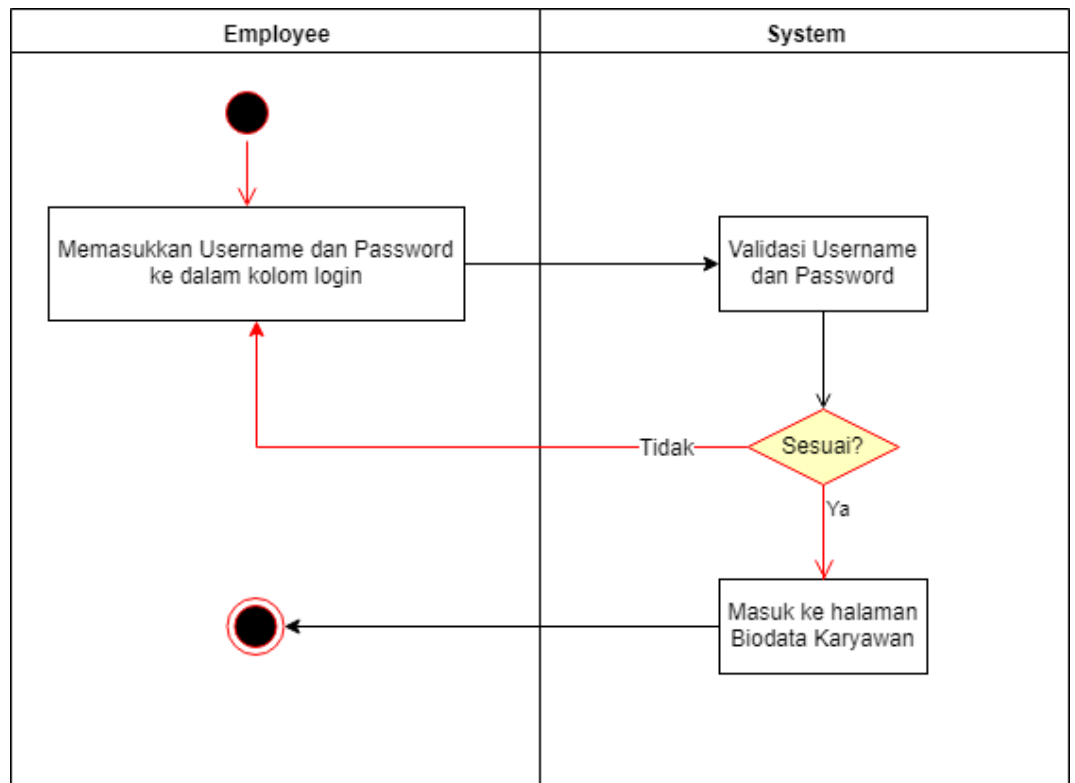
Activity Diagram Hubungi yang nantinya apabila *employee seeker* sudah merasa *employee* yang dipilih sesuai dengan kriteria, maka *employee seeker* dapat langsung menghubungi *employee* dengan cara menekan tombol hubungi, dapat digambarkan seperti dibawah ini.



Gambar 1-6 Hubungi

6. Activity Diagram Login

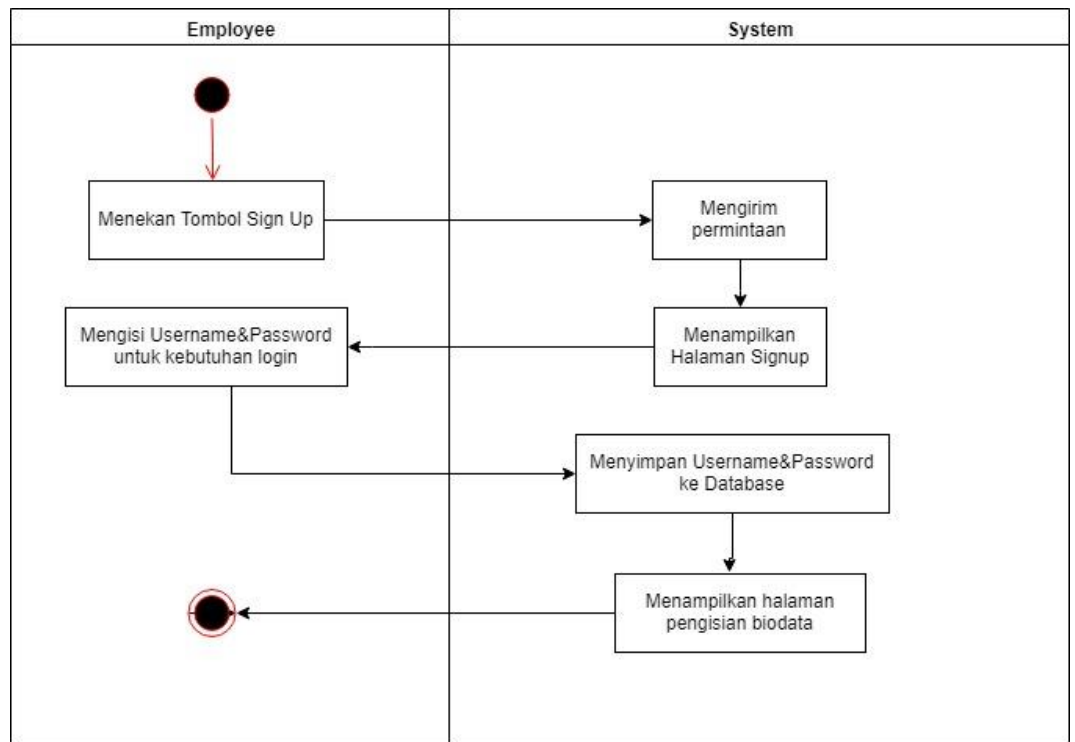
Activity Diagram Login yang nantinya *employee* diharuskan untuk memasukkan *username* dan *password*, tetapi dengan catatan bahwa *employee* sudah melakukan *registrasi* terlebih dahulu diawal, dapat digambarkan seperti dibawah ini.



Gambar 1-7 Login

7. Activity Diagram Sign Up

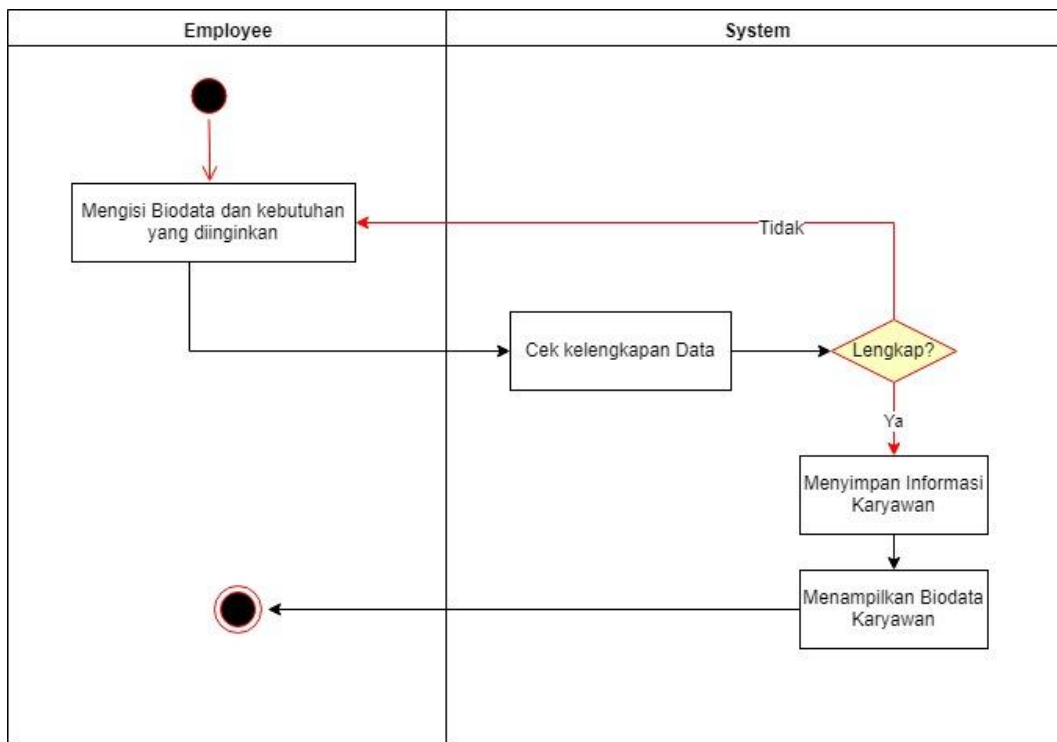
Activity Diagram Sign Up yang nantinya *employee* yang ingin mendapatkan pekerjaan paruh waktu, diharuskan untuk mendaftar terlebih dahulu dengan memasukkan *username*, *password* dan *email*, dapat digambarkan dibawah ini.



Gambar 1-8 Sign Up

8. Activity Diagram Save Data

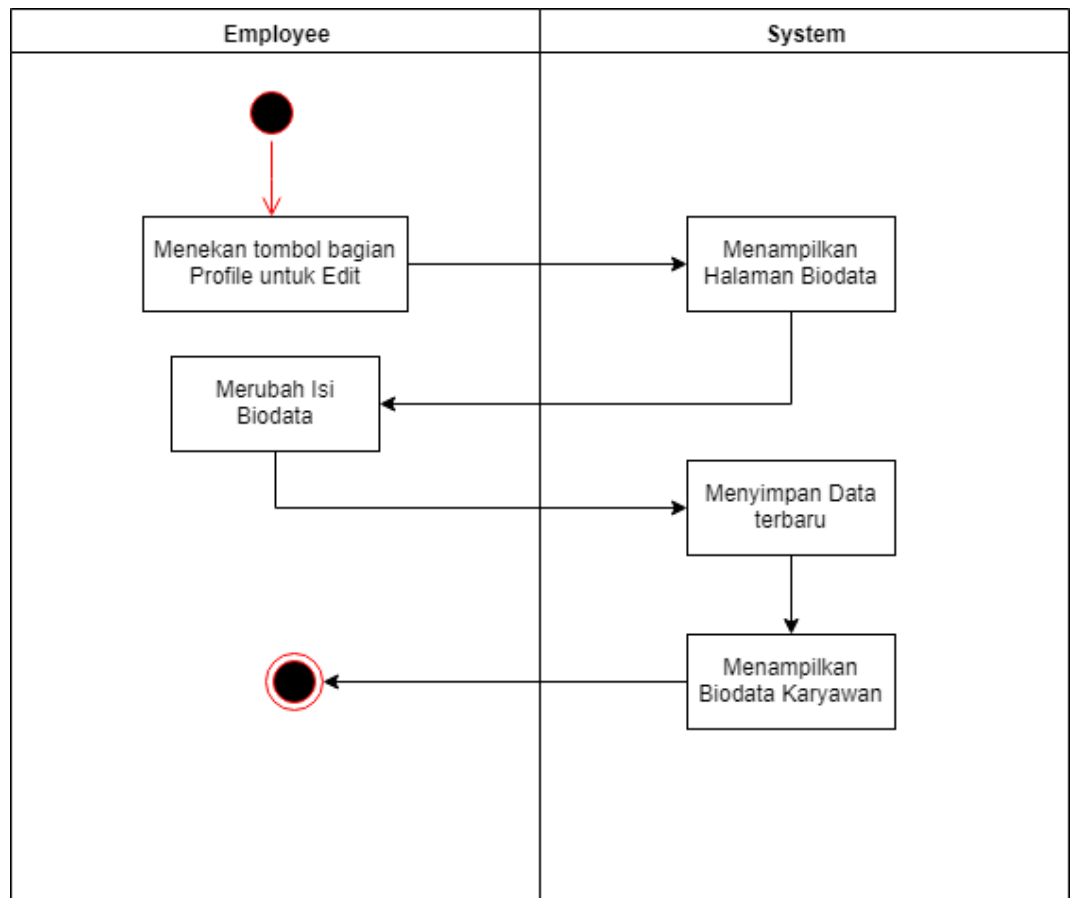
Activity Diagram Save Data yang nantinya menyimpan seluruh biodata *employee* ke dalam *database*, dapat digambarkan seperti dibawah ini.



Gambar 1-9 Save Data

9. Activity Diagram Pengelolaan Data Employee

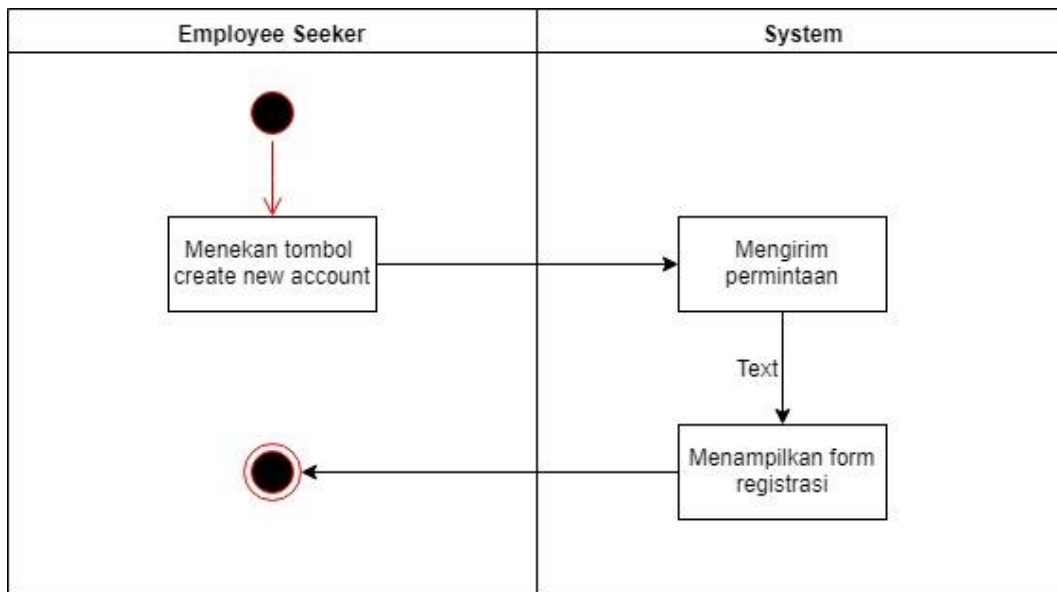
Activity Diagram Biodata Employee yang nantinya ketika employee sudah selesai melakukan pendaftaran, maka dilanjutkan untuk login setelah itu akan muncul biodata diri dari employee, apabila terjadi data yang salah maka dapat di edit dengan cara mengetuk profil yang sudah disediakan diatas, dapat digambarkan seperti dibawah ini.



Gambar 1-10 Pengelolaan Data Employee

10. Activity Diagram Membuat Akun Baru

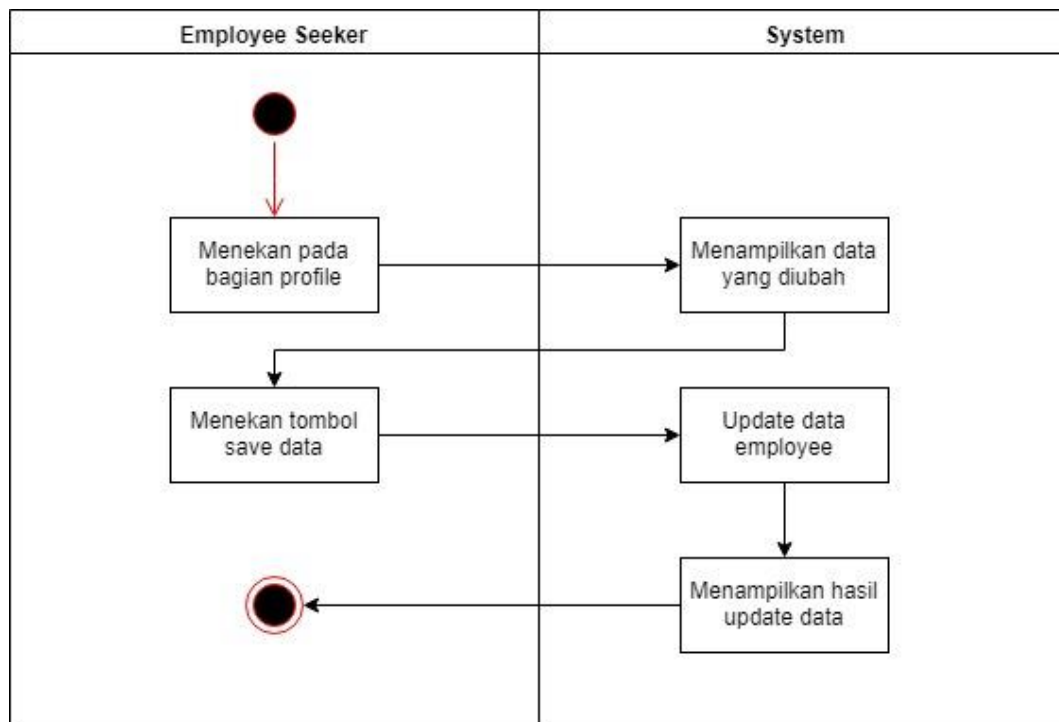
Activity Diagram Membuat Akun Baru nantinya ketika *employee seeker* masuk ke menu *login* tetapi belum melakukan *registrasi* terlebih dahulu, dapat langsung menekan tombol *create new account*, dapat digambarkan seperti dibawah ini.



Gambar 1-11 Membuat Akun Baru

11. Activity Diagram Edit Data

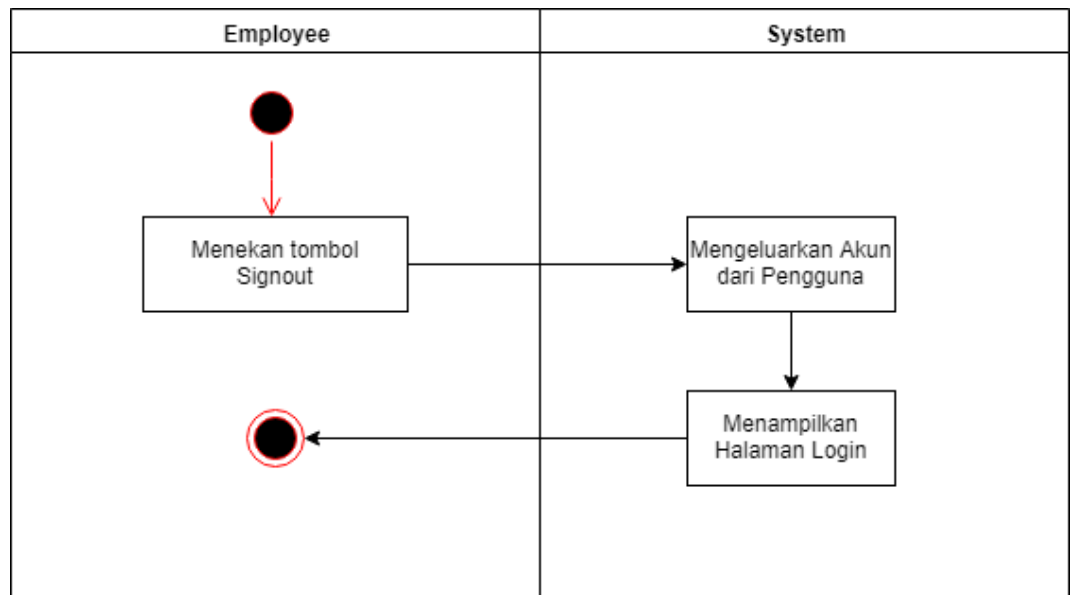
Activity Diagram Edit Data nantinya ketika *employee* ingin merubah biodata diri, dapat mengetuk *profil* pada biodata *employee*, dapat digambarkan seperti dibawah ini.



Gambar 1-12 Edit Data

12. Activity Diagram Sign Out

Activity Diagram Sign Out yang nantinya ketika *employee* sudah melakukan *login* dan sudah merasa bahwa data yang di *input* sudah benar, maka *employee* dapat keluar dari akun dengan cara menekan tombol *sign out*, dapat digambarkan seperti dibawah ini.



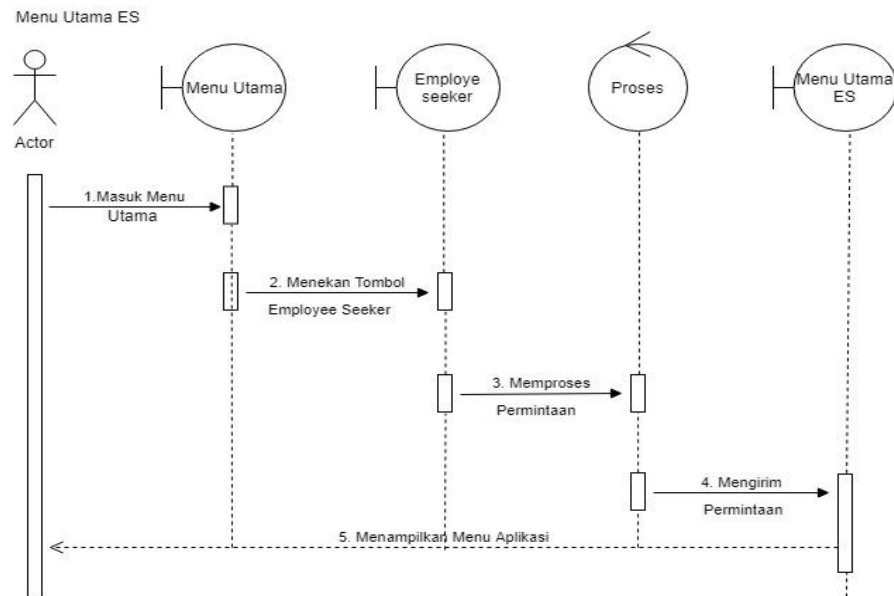
Gambar 1-13 Sign Out

3.1.2.7 Sequence Diagram

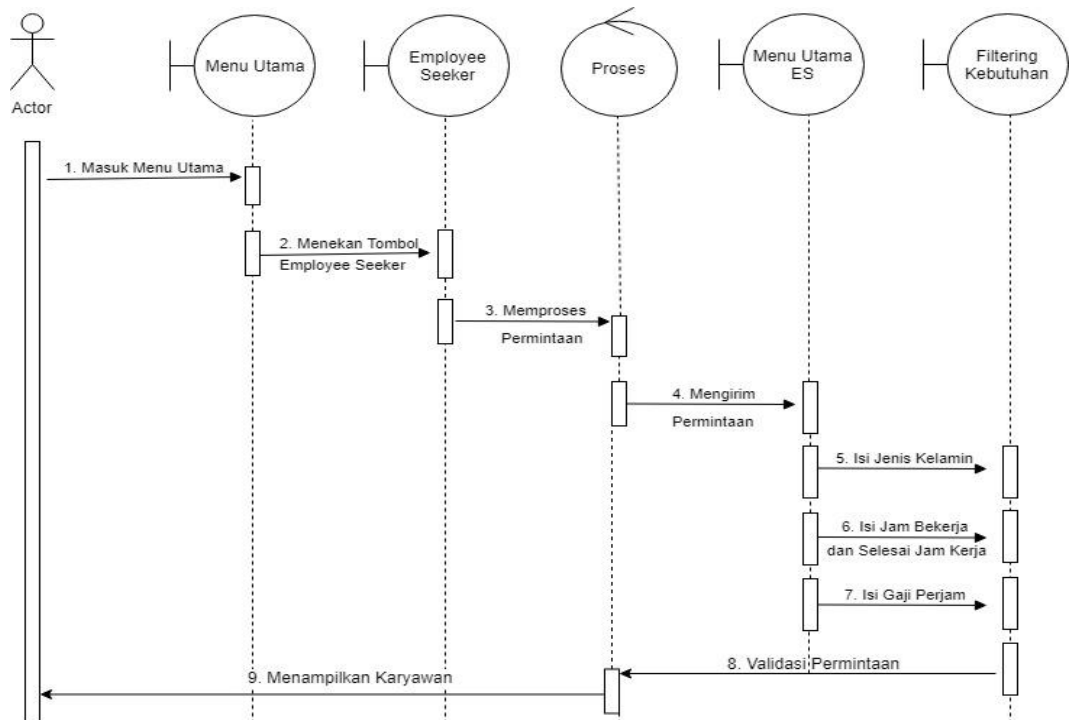
sequence diagram ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, berikut beberapa *sequence diagram* yang ada pada aplikasi dan dapat dilihat pada gambar dibawah ini.

1. Sequence Diagram Menu Utama ES

Gambar 1-14 Menu Utama ES

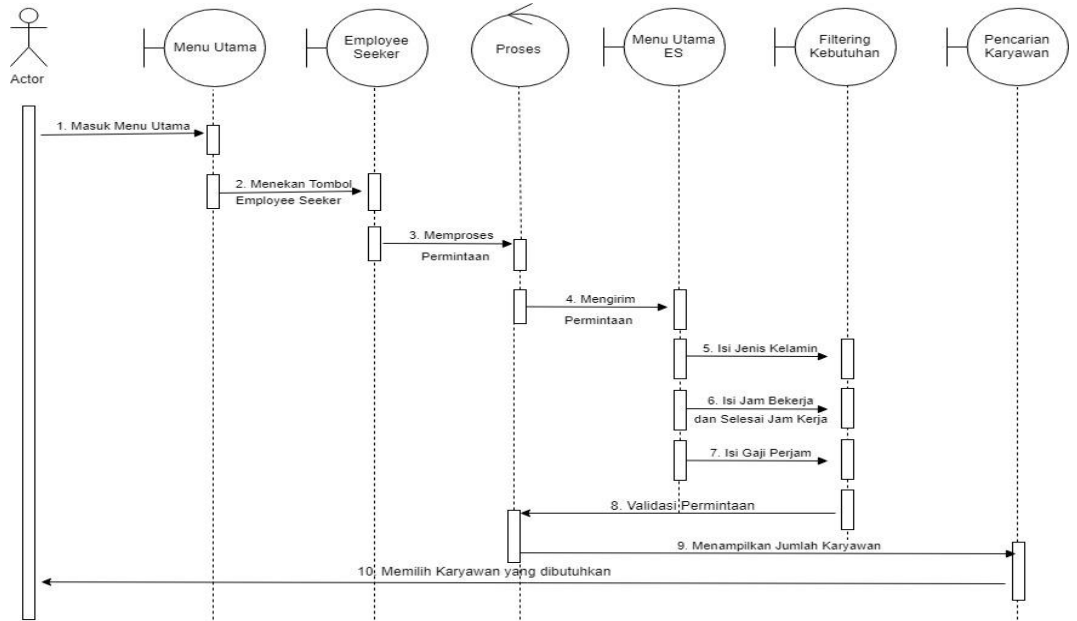


2. Sequence Diagram Filtering Kebutuhan



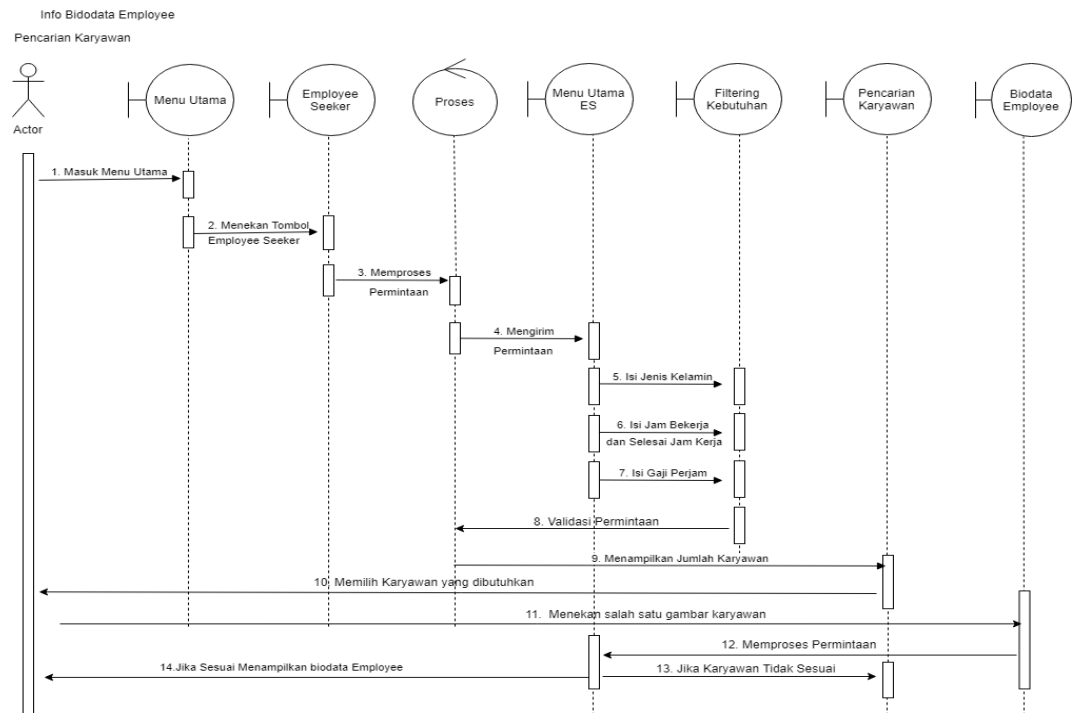
Gambar 1-15 Filtering Kebutuhan

3. Sequence Diagram Pencarian karyawan



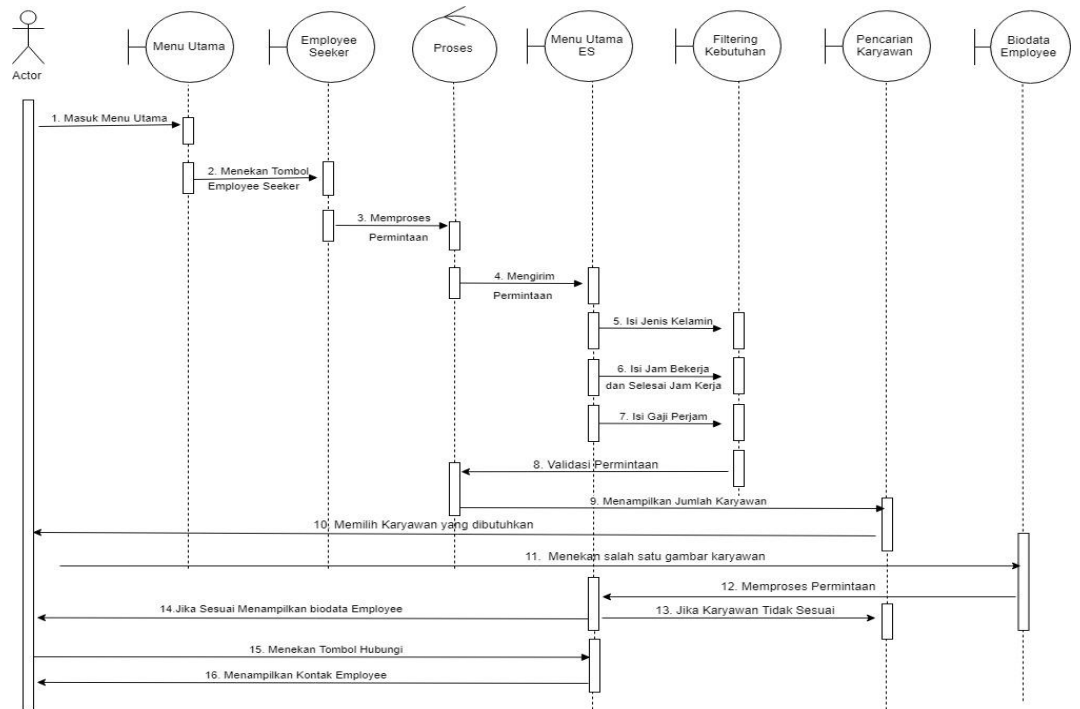
Gambar 1-16 Pencarian Karyawan

4. Sequence Diagram Info Biodata Employee



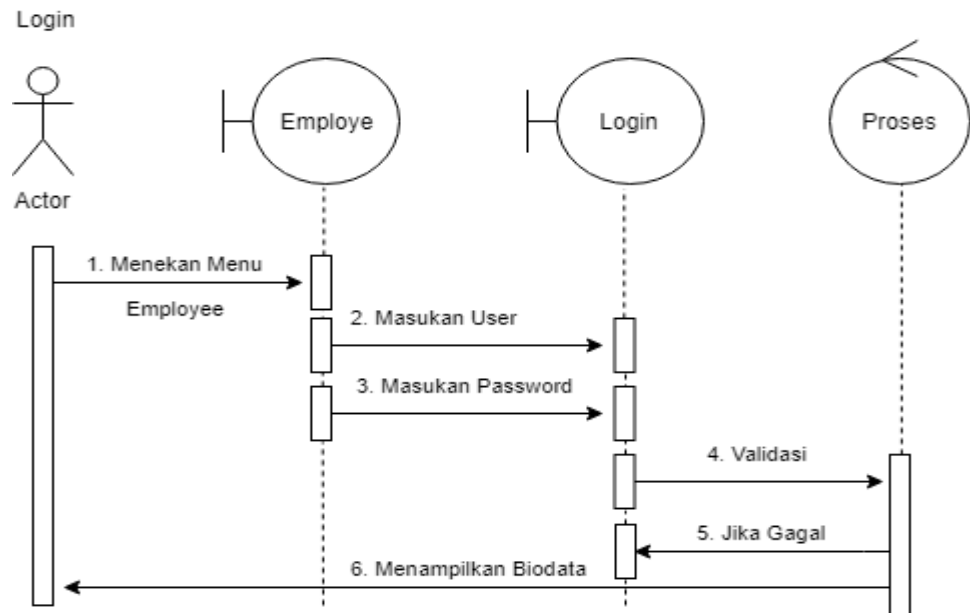
Gambar 1-17 Biodata Employee

5. Sequence Diagram Hubungi



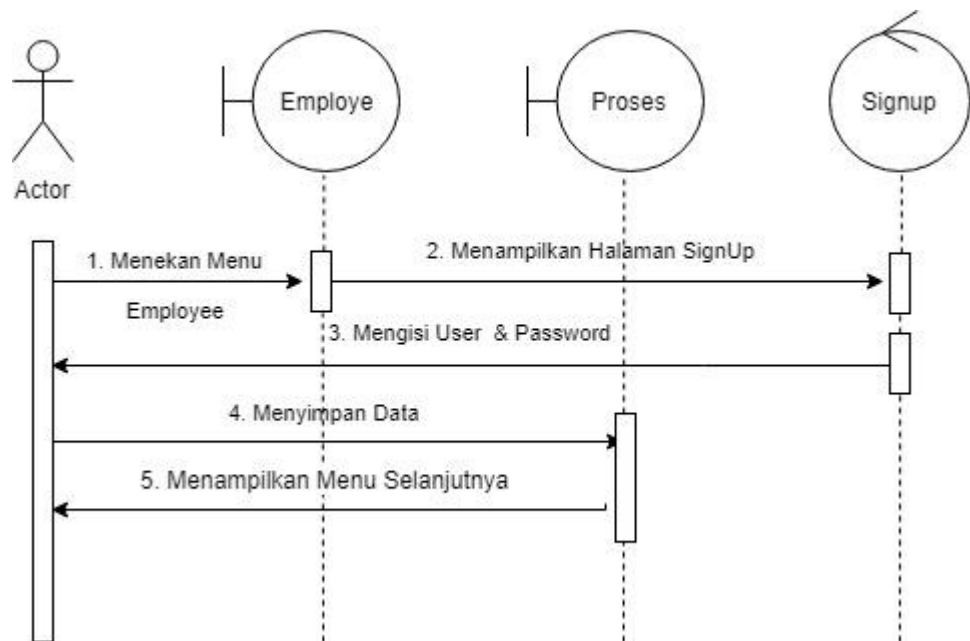
Gambar 1-18 Hubungi

6. Sequence Diagram Login



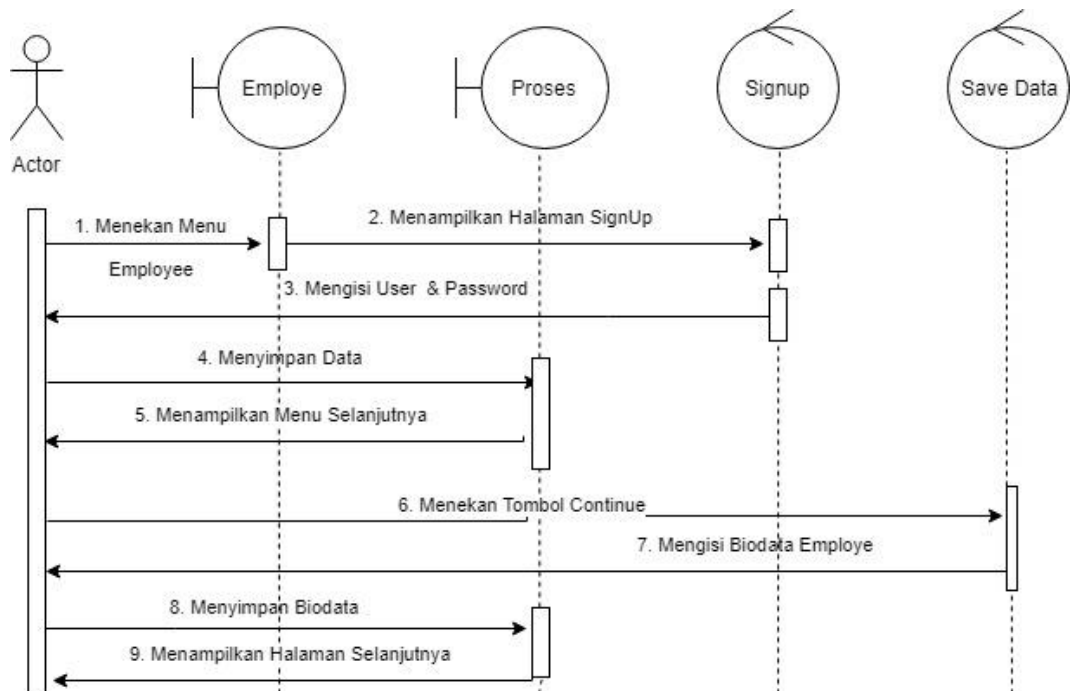
Gambar 1-19 Login

7. Sequence Diagram Sign Up



Gambar 1-20 Sign Up

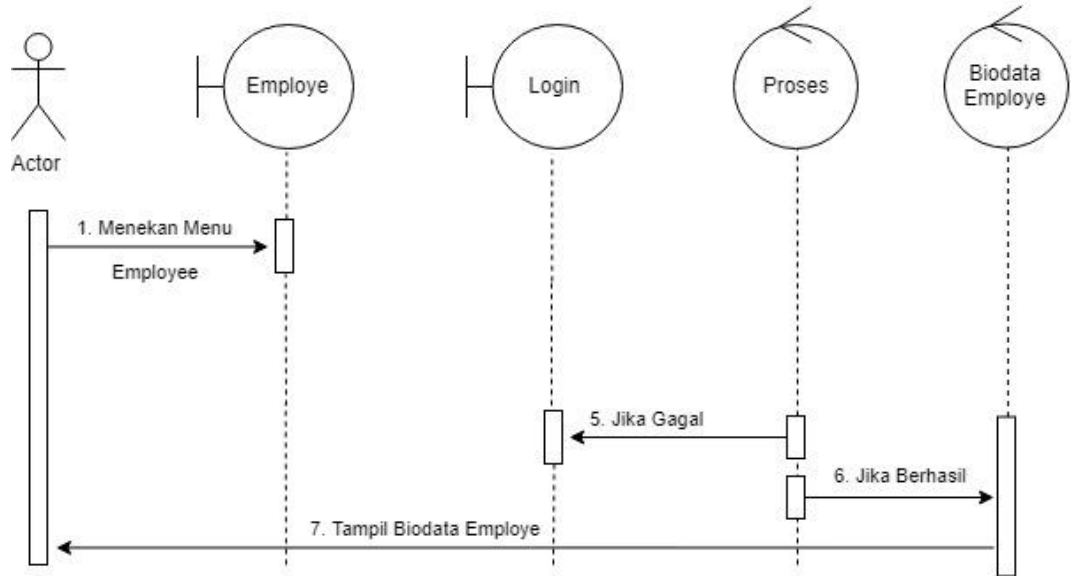
8. Sequence Diagram Save Data



Gambar 1-21 Save Data

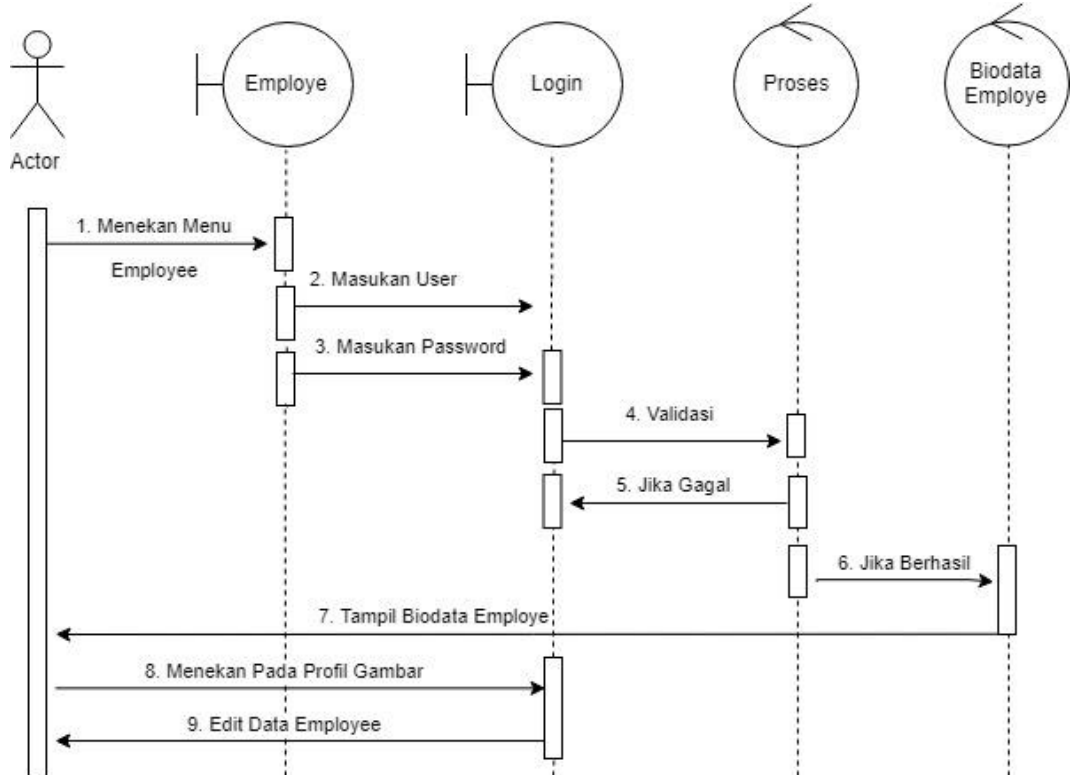
9. Sequence Diagram Biodata Employee

Sequence Biodata employee



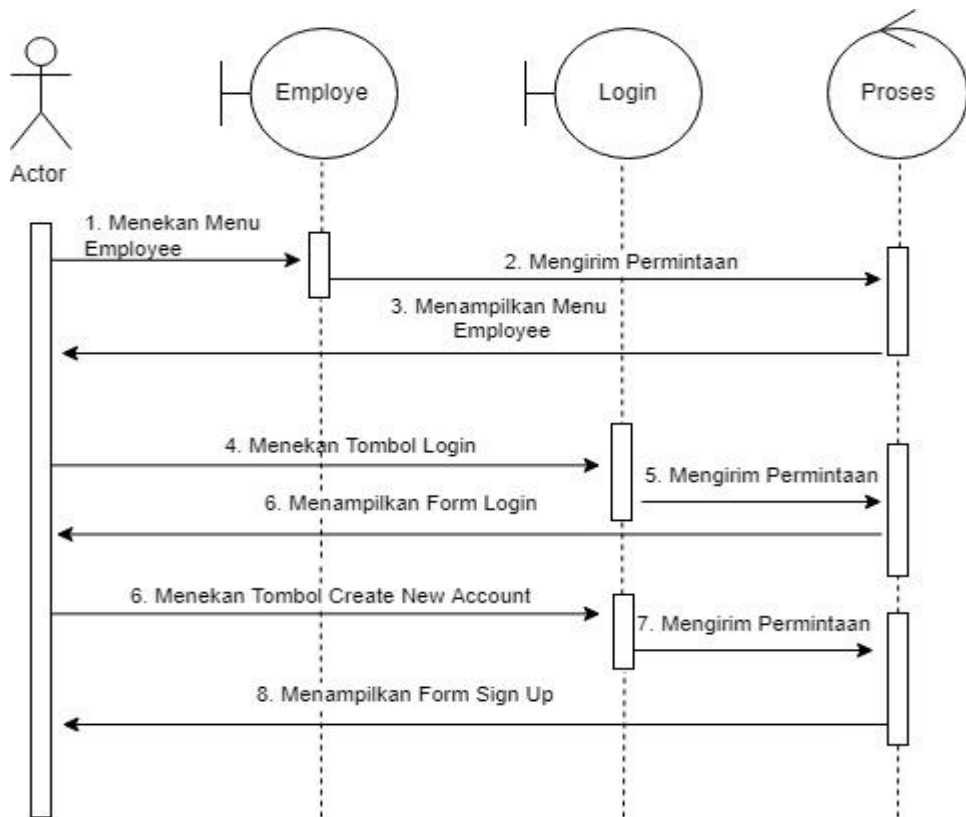
Gambar 1-22 Biodata Employee

10. Sequence Diagram Edit Data



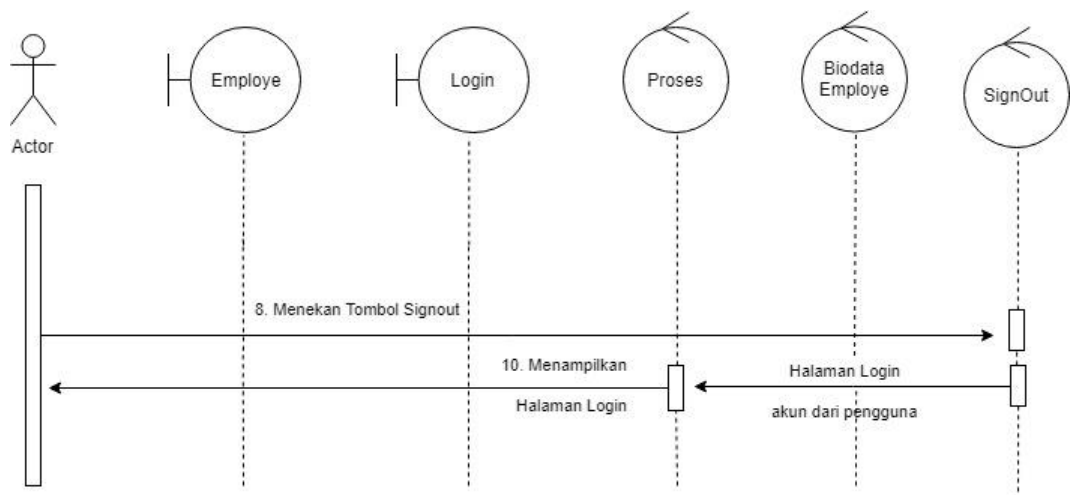
Gambar 1-23 Edit Data

11. Sequence Diagram Membuat Akun Baru



Gambar 1-24 Membuat Akun Baru

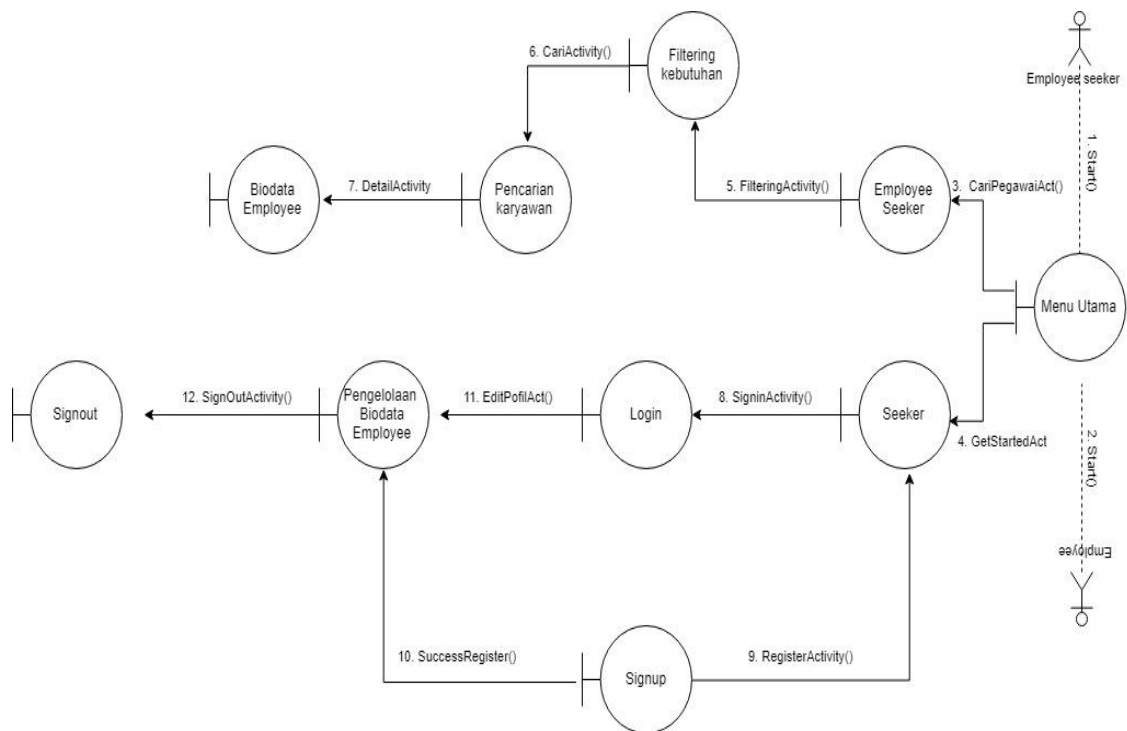
12. Sequence Diagram Sign Out



Gambar 1-25 Sign Out

3.1.2.8 Collaboration Diagram

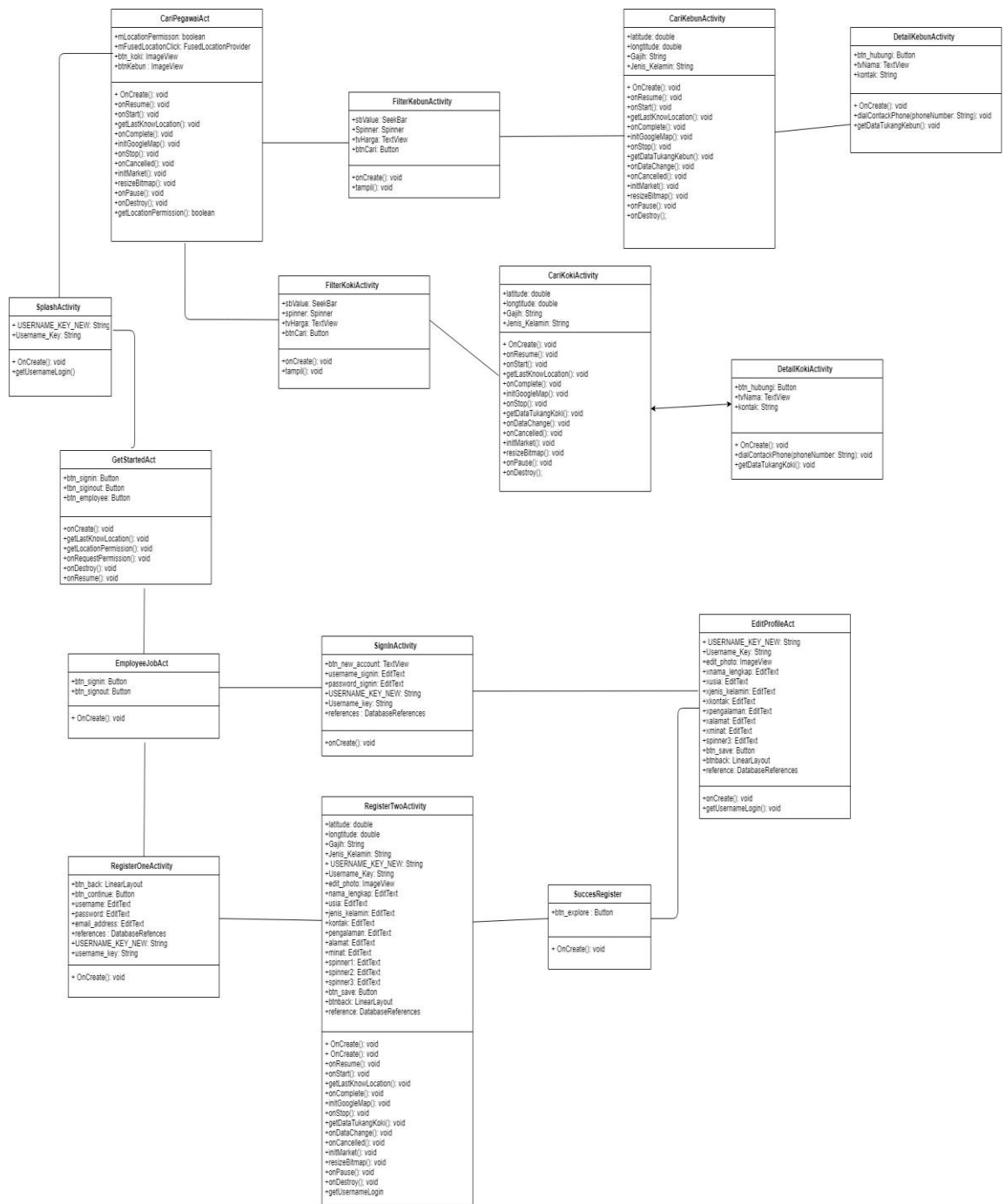
Collaboration diagram adalah diagram yang mengelompokkan pesan pada kumpulan diagram sekuen menjadi sebuah diagram. Dalam diagram tersebut terdapat method yang dijalankan antara objek yang satu dan objek lainnya. Di diagram kolaborasi ini, objek harus melakukan *sinkronisasi* pesan dengan serangkaian pesan-pesan lainnya, *Collaboration Diagram* dapat digambarkan seperti dibawah ini



Gambar 1-26 Collaboration Diagram Aplikasi

3.1.2.9 Class Diagram

Class diagram adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki 3 bagian utama yaitu *attribute*, *operation*, dan *name*. kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem.



Gambar 1-27 Class Diagram

3.1.6 Analisis Kebutuhan Non Fungsional

Adapun yang menjadi isi dari kebutuhan non fungsional adalah:

3.1.2.10 Analisis Perangkat Lunak

Analisis kebutuhan perangkat lunak menunjukkan kebutuhan perangkat lunak yang direkomendasikan pada tahap implementasi. Kebutuhan perangkat lunak memiliki spesifikasi minimum seperti pada *Tabel* dibawah ini:

Tabel 1-1 Analisis Perangkat Lunak

No	Kebutuhan Perangkat Lunak	Spesifikasi
1	Oprating System	Android Lollipop 5.0
2	Memori	1 GB
3	RAM	2 GB

3.2 Perancangan Sistem

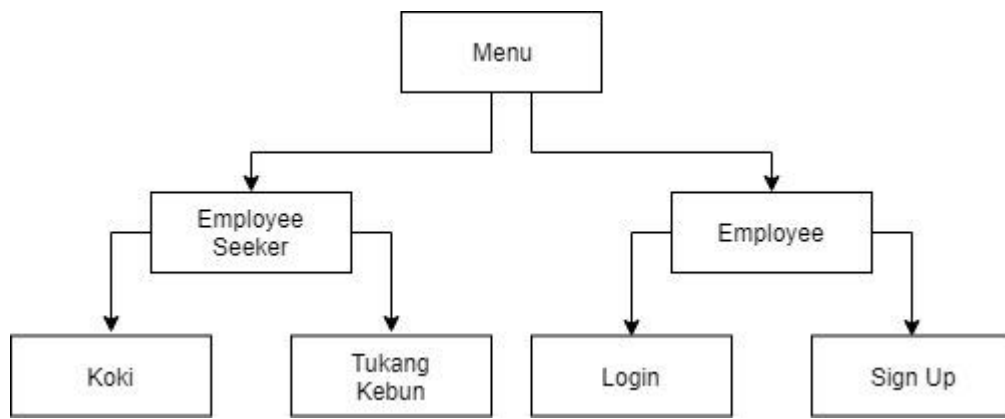
Perancangan Sistem dibuat untuk mempermudah dalam pengerjaan aplikasi, dimana dalam perancangan sistem terdapat perancangan arsitektural , perancangan data, perancangan antarmuka dan perancangan prosedural.

3.1.7 Perancangan Arsitektural

Perancangan arsitektural merupakan susunan konsep dalam pembangunan aplikasi, mulai dari awal sampai akhir pembangunan. Perancangan arsitektural juga dapat memudahkan dalam pembangunan, karena nantinya dalam perancangan ini kita membuat tampilan *menu* maupun isi dari aplikasi yang akan dibangun.

3.1.2.11 Perancangan Struktur Menu

Perancangan struktur menu merupakan gambaran dari aplikasi yang akan dibuat, dengan tujuan untuk membantu dan memudahkan para pengguna untuk mengetahui menu apa saja yang akan dibuat dan cara penggunaannya. Dibawah ini merupakan gambar struktur menu dari aplikasi yang akan dibangun.



Gambar 2-1 Struktur Menu

3.1.8 Perancangan Data

Perancangan data merupakan proses untuk menentukan isi dan kebutuhan data yang dibutuhkan untuk mendukung berbagai rancangan sistem, nantinya hasil data didapatkan untuk mempermudah penulis merancang sistem yang akan dibangun.

Tabel 2-1 Perancangan Data

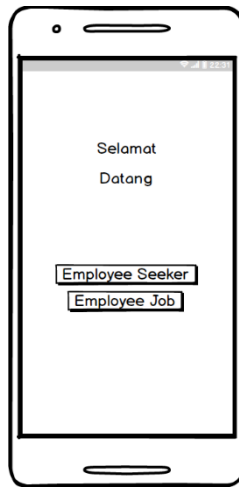
No	Field Name	Type	Description
1	Lokasi	Map	Digunakan untuk mengetahui titik keberadaan dari <i>employee</i>
2	Gambar-gambar	Img	Gambar yang digunakan sebagai kebutuhan dalam aplikasi

3	<i>Profile</i>	<i>Img</i>	Gambar yang digunakan untuk mengetahui wajah <i>employee</i>
4	Tampilan <i>Employee Seeker</i>	<i>Teks</i>	Digunakan untuk menampilkan isi dari <i>employee seeker</i> yang nantinya untuk mencari <i>employee</i> sesuai yang diinginkan.
5	Tampilan <i>Employee</i>	<i>Teks</i>	Digunakan untuk menampilkan informasi terkait <i>employee</i> itu sendiri
6	Tombol <i>Menu</i>	<i>Button</i>	Tombol di <i>menu</i> utama yang akan menghubungkan dengan fungsi lain

3.1.9 Perancangan Antarmuka

Perancangan antarmuka dibuat untuk mewakili ataupun memberikan gambaran terkait aplikasi yang akan dibangun, yang nantinya dari antarmuka tersebut dapat mengetahui fungsi-fungsi apa saja yang digunakan dan manfaatnya seperti apa. Berikut adalah perancangan antarmuka yang akan dibangun.

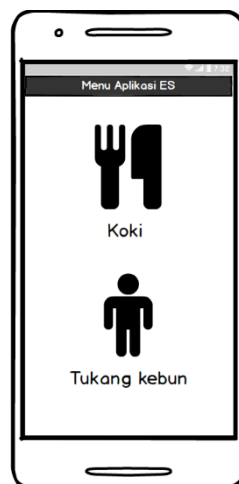
1. Menu Utama Aplikasi



Gambar 2-2 Menu Utama Aplikasi

Pada gambar di atas menggambarkan tampilan awal atau *menu* utama ketika memasukin aplikasi ini, terdapat dua menu didalamnya yaitu *employee seeker* dan *employee job*, apabila kita sebagai penyedia jasa ingin mencari karyawan, maka diharuskan untuk memilih menu *employee seeker*, jika kita seorang karyawan yang ingin mendaftarkan pekerjaan paruh waktu, kita dapat masuk ke menu *employee job*

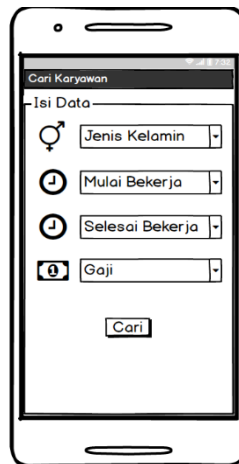
2. Menu Utama *Employee Seeker*



Gambar 2-3 Menu Utama Employee Seeker

Pada gambar di atas menggambarkan tampilan awal ketika masuk kedalam menu *employee seeker*, nantinya terdapat dua menu utama yaitu koki, dan tukang kebun.

3. *Filtering* Kebutuhan



Gambar 2-4 *Filtering* Kebutuhan

Pada tampilan *employee seeker* berikutnya seperti pada gambar diatas, nantinya *employee seeker* mengisi data-data penting yang dibutuhkan, mulai dari memilih jenis kelamin, jam mulai bekerja, jam selesai bekerja dan yang terakhir gaji sesuai dengan yang kita inginkan perjamnya.

4. Pencarian Karyawan



Gambar 2-5 Pencarian Karyawan

Pada gambar diatas menunjukkan titik/*employee* yang tersedia dari hasil pencarian pada sistem sesuai dengan yang kita butuhkan, apabila karyawan yang tidak sesuai dengan kriteria kita maka tidak akan muncul dalam *maps* tersebut.

5. Info Biodata *Employee*



Gambar 2-6 Info Biodata *Employee*

Pada gambar diatas memberikan informasi berupa biodata dan kontak person, nantinya *employee seeker* bisa mengetahui pengalaman kerja dari *employee*, apabila sesuai dengan yang dicari dan diinginkan maka dapat langsung menghubungi dari kontak yang disediakan.

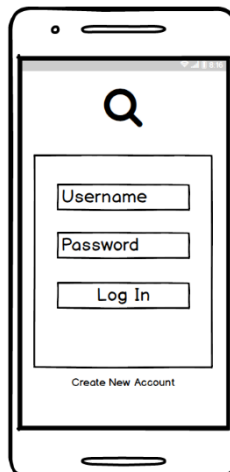
6. Menu Utama *Employee*



Gambar 2-7 Menu Utama Employee

Pada gambar di atas menggambarkan ketika kita memilih menu *employee* maka akan muncul dua menu, diantaranya *login* dan *sign up*, untuk karyawan yang belum mendaftar diharapkan memilih menu *Sign Up*, dan apabila sudah mendaftar dapat masuk ke *Log In*

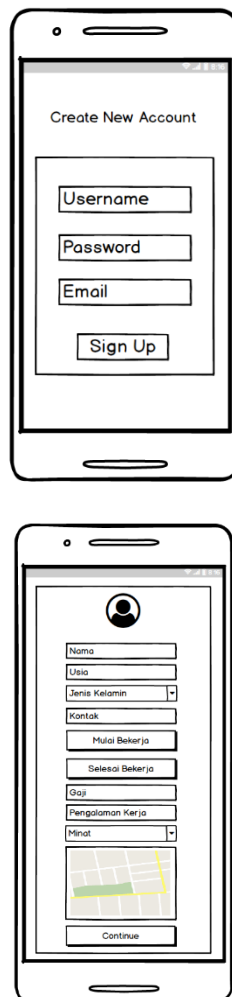
7. *Login*



Gambar 2-8 Login

Pada gambar di atas menggambarkan ketika karyawan sudah melakukan *registrasi*, maka diharuskan untuk melakukan *login* agar mengetahui lebih detail informasi dari karyawan itu sendiri

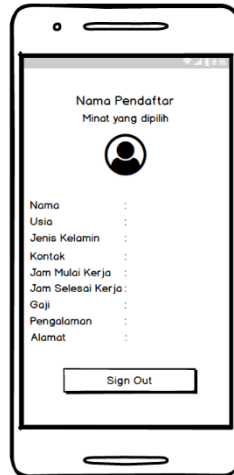
8. *Sign Up*



Gambar 2-9 *Sign Up*

Pada gambar di diatas menunjukkan format yang dibutuhkan ketika ingin mendaftar pada aplikasi ini, nantinya para *employee* yang ingin mendaftar dapat mengisi sesuai list di atas.

9. Biodata *Employee*

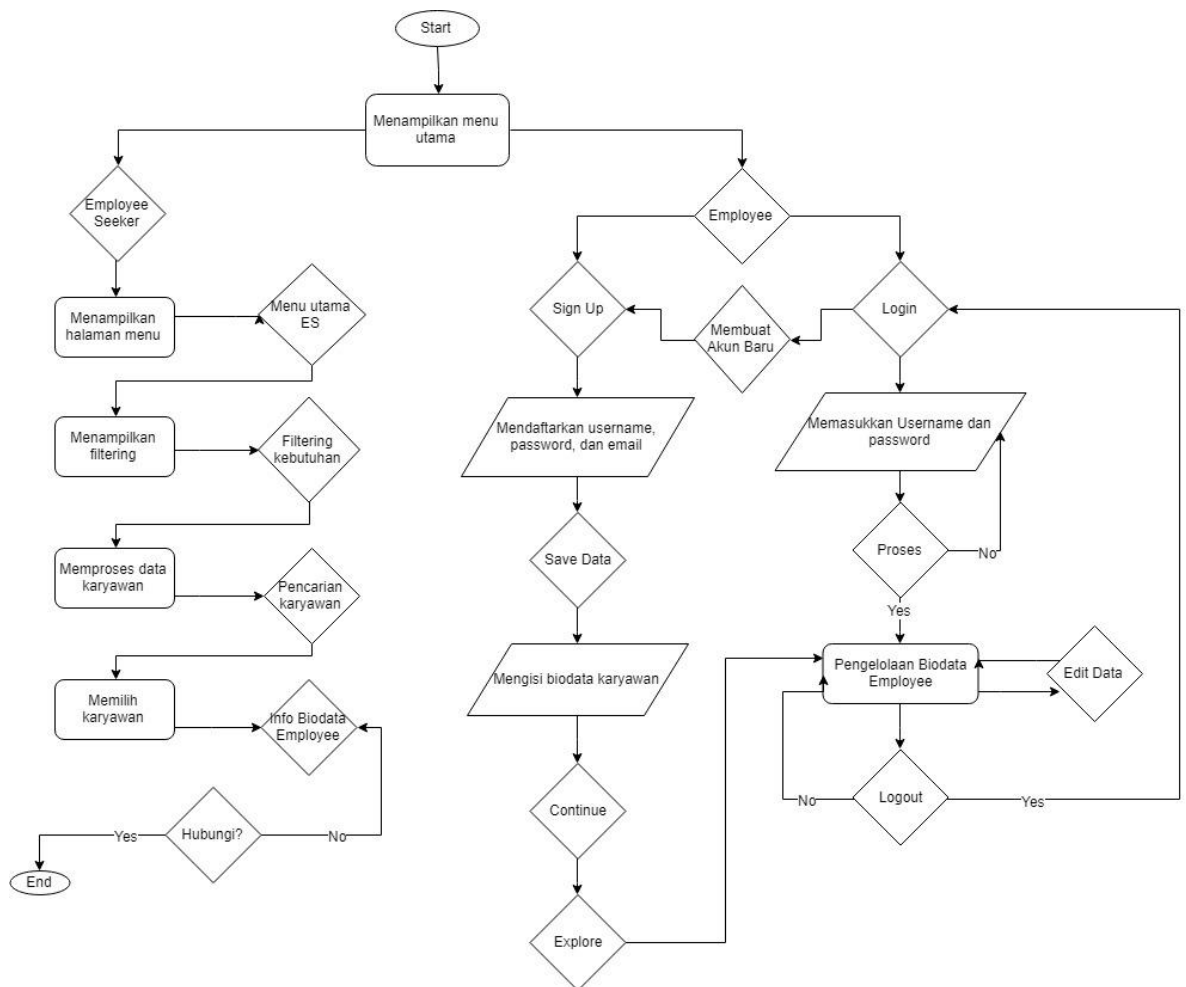


Gambar 2-10 Biodata *Employee*

Pada gambar diatas menunjukkan informasi akhir dari *employee*, nantinya biodata itu lah yang akan dilihat dan menjadi parameter untuk *employee seeker* memilih para *employee-employee* yang ada.

3.1.10 Perancangan Prosedural (Flowchart)

Perancangan *Prosedural* merupakan perancangan yang dibuat untuk menetapkan algoritma yang akan ditetapkan di dalam *system*. Perancangan *procedural* sendiri digambarkan dalam *flowchart*, yang digambarkan dibawah ini:



Gambar 2-11 Perancangan Prosedural (Flowchart)

Keterangan dari *flowchart* diatas akan dijelaskan dalam *tabel* dibawah ini :

Tabel 2-2 Alur FLOWchart

Keterangan
Aplikasi Akan dijalankan.
Aplikasi akan memunculkan <i>menu</i> utama, terdapat <i>menu Employee Seeker</i> dan <i>Employee</i>
<i>Menu employee seeker</i> hanya digunakan untuk para <i>employee seeker</i> yang akan mencari <i>employee</i> , nantinya para <i>employee seeker</i> memilih <i>menu employee seeker</i> .

Terdapat menu *koki* dan tukang kebun, nantinya *employee seeker* memilih sesuai kebutuhannya.

Setelah itu dapat mengisi data untuk *filtering employee* yang dibutuhkan, mulai dari jenis kelamin, waktu mulai bekerja, waktu selesai bekerja, gaji, dan jarak maksimal dari lokasi yang diinginkan.

Nantinya akan muncul *employee-employee* yang sesuai dengan yang kita butuhkan, dari sekian banyak *employee* yang masuk dalam kategori, kita dapat memilih salah satunya dengan cara menekan pada gambar, lalu akan muncul informasi berupa nama dan kontak, setelah itu di klik kembali apabila ingin mengetahui lebih jauh.

Akan muncul biodata *employee* yang tadinya dipilih, informasi *employee* akan lebih *detail* didalamnya termasuk pengalaman kerja sebelumnya, yang itu akan menjadi tolak ukur untuk memilih karyawan tersebut.

Setelah sudah merasa cocok kita dapat langsung menghubungi *employee* dengan cara menekan tombol *employee* dipaling bawah pada biodata *employee*

Pada menu *employee* terdapat dua menu, yaitu *login* dan *sign up*

Pada menu *login* nantinya diarahkan untuk mengisi *username* dan *password* yang telah didaftarkan sebelumnya

setelah melakukan login dapat muncul informasi data *employee* yang sudah diisi pada *form registrasi*

Pada menu *login* juga terdapat pilihan untuk membuat akun baru, ketika menekan pilihan tersebut maka nantinya kita diarahkan untuk ke *form registrasi*

Pada menu *sign up* nantinya diarahkan untuk mendaftarkan *username*, *password*, dan *email*, lalu menekan tombol *sign up*

Setelah itu diarahkan untuk mengisi biodata *employee* yang ingin mendaftar, lalu tekan tombol *save data*

Setelah melakukan *registrasi* maka akan langsung muncul biodata yang sudah diisi sebelumnya

Ketika informasi data diri sudah muncul, kita dapat mengubah data yang sudah ada dengan cara menekan para *profil*, lalu data dapat diubah dan tekan *save profile*

Apabila sudah melakukan *edit* data, kita dapat keluar dari *akun* dengan cara menekan tombol *Sign Out*, maka setelah itu akan diarahkan ke menu *login* kembali